AppCritique Security Report

Xumo
Core Report
Date Reviewed: 2019-12-01
Version Name: 1.1

Platform	Package Name	Version Code	SHA-256 Hash
Android	com.xumo.xumo.tv	34	f8a5b7475c4ef2b082aeb62368914d2e fb84ab540be8189f8cb62fa774fb6613

Certificate Information

Name	Explanation
Owner Name	Jiro Egawa
Organization	Xumo
Organizational Unit	LLC
Location	Irvine, CA, US
Validity	October 05, 2016 through September 29, 2041

Total number of flaws or potential flaws found: 13

Functionality

Functionality Present

Name	Explanation
Keychain	This app stores data in the device Keystore.
Networking	This app connects to the internet and requests the following network related permission(s): android.permission.ACCESS_NETWORK_STATE, android.permission.ACCESS_WIFI_STATE, and android.permission.INTERNET.
Sensors	This app uses device sensors.

Functionality Not Found

Audio, Bluetooth, Camera, Calendar, Contacts, Device Administrator, Fingerprint, Google Cloud Messaging, Geofencing, Health Data, Infrared LED, In-App Purchases, Location, Maps, Microphone, NFC, Payment Services, Photos, SMS, Telephony, USB Devices

Imported Libraries

Social Networks

Facebook	✘	Flickr	✘	Foursquare	✘
Google+	✘	Instagram	✘	LinkedIn	✘
Pinterest	✘	Tumblr	✘	Twitter	✔
Yelp	✘				

Analytics Networks

Adjust	✘	AdobeMarketingCloud	✘	AmazonAnalytics	✘
AmazonInsights	✘	Amplitude	✘	AppBoy	✘
Applause	✘	Appsflyer	✘	Apptimize	✘
Apsalar	✘	Branch	✘	Countly	✘
Flurry	✘	GoogleAnalytics	✘	Kochava	✘
Localytics	✘	Mixpanel	✘	MobileAppTracking	✘
NewRelic	✘	Quantcast	✘	Tapstream	✘
Vessel	✘	Webtrends	✘		

Advertising Networks

AdColony	✘	Adfalcon	✘	Admob	✘
AmazonAds	✘	Amobee	✘	AppBrain	✘
AppLovin	✘	Appnexus	✘	Axonix	✘
Chartboost	✘	DoubleClick	✔	FlurryAds	✘

FusePowered	✘	Fyber	✘	IMOBI	✘
Inneractive	✘	Kiip	✘	Liquid	✘
Madvertise	✘	MdotM	✘	mMedia	✘
Mobfox	✘	MobPartner	✘	NativeX	✘
RevMob	✘	SessionM	✘	Smaato	✘
Tapjoy	✘				

Cloud Storage

Box	✘	Cloud Drive	✘	Dropbox	✘
GoogleDrive	✘	MediaFire	✘	OneDrive	✘

Developer Tools

aChartEngine	✘	ActiveAndroid	✘	aFileChooser	✘
AmazonDeviceMessaging	✘	AndroidLogger	✘	Annotations	✘
Answers	✘	Appsee	✘	AsyncHttp	✘
Beacon	✘	BitmapCache	✘	ButterKnife	✘
Digits	✘	Fabric	✔	Firefly	✘
Parse	✘	Paypal	✘	PhoneGap	✘
PubNub	✘	Retrofit	✘	Spotify	✘
Stripe	✘	UniversalImageLoader	✘		

Permissions Requested

Permission Name	Is Used	Protection Level
android.permission.ACCESS_NETWORK_STATE		**Normal**
android.permission.ACCESS_WIFI_STATE		**Normal**
android.permission.INTERNET	✔	**Normal**
android.permission.RECEIVE_BOOT_COMPLETED	✔	**Normal**
android.permission.WAKE_LOCK	✔	**Normal**
android.permission.WRITE_EXTERNAL_STORAGE	✳	**Dangerous**
com.google.android.c2dm.permission.RECEIVE	✳	
com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE	✳	
com.xumo.xumo.c2dm.permission.C2D_MESSAGE	✳	**Signature**

✳ Unable to detect use of this permission.

Security Checks

Checks Conducted

Check	Result	Explanation
Accesses External Storage	**Present**	The app accesses the external storage directory, also referred to as the SDCard. External storage can be accessed by any app on a device with the READ/WRITE_EXTERNAL_STORAGE permission. It is therefore recommended not to store any sensitive information in external storage. External storage access is found in the following methods and classes: •androidx.core.content.ContextCompat.getExternalCacheDirs(Context) •androidx.core.content.ContextCompat.getExternalFilesDirs(Context, String) •androidx.core.content.ContextCompat.getObbDirs(Context) •androidx.core.content.FileProvider.parsePathStrategy(Context, String) •androidx.core.os.EnvironmentCompat.getStorageState(File) •com.xumo.xumo.application.XumoApplication.getDownloadDirectory() •io.fabric.sdk.android.FabricContext.getExternalCacheDir() •io.fabric.sdk.android.FabricContext.getExternalFilesDir(String) •io.fabric.sdk.android.services.persistence.FileStoreImpl.getExternalCacheDir() •io.fabric.sdk.android.services.persistence.FileStoreImpl.getExternalFilesDir() OWASP: 2016-M2-Insecure Data Storage NIAP: FDP_DAR_EXT.1.1
Accesses Unique Identifiers	**Not Present**	The app does not access any unique identifiers.
Activities Accessible to Other Apps	**Not Present**	No activities are exported, or access to all activities is restricted by use of permissions. OWASP: 2016-M1-Improper Platform Usage NIAP: FCS_NET_EXT.1.1

Allows Data to be Backed up and Restored	**Present**	The app allows backup of its data. A malicious actor with physical access to the device could get access to sensitive data by retrieving private files, databases, shared preferences files, caches or libraries within the app. OWASP: 2016-M1-Improper Platform Usage
App is Debuggable	**Not Present**	The app is not debuggable. This protects the app against reverse engineering and the execution of arbitrary code. OWASP: 2016-M10-Extraneous Functionality
Contains Hard-coded Cryptographic Key	**Not Present**	No hard-coded cryptographic keys were found in the app. OWASP: 2016-M5-Insufficient Cryptography; 2016-M9-Reverse Engineering
Contains HostnameVerifier That Accepts All Hostnames	**Not Present**	No weak HostnameVerifiers are found. OWASP: 2016-M3-Insecure Communication NIAP: FCS_TLSC_EXT.1.2
Contains Native Code	**Present**	The app loads native code libraries. Native code does not have the same security protections as Java, and is vulnerable to buffer overflows, use after free errors, and off-by-one errors. Native code can also be loaded from untrusted sources, such as a shared directory or the network. OWASP: 2016-M7-Client Code Quality NIAP: FPT_AEX_EXT.1.5
Contains Potential Hard-coded Password	**Not Present**	No hard-coded passwords were found in the app. OWASP: 2016-M9-Reverse Engineering
Contains Potential SQL Injection	**Not Present**	No potential SQL injection vulnerabilities were found. OWASP: 2016-M7-Client Code Quality
Contains Reflection Code	**Present**	The app contains Java reflection code. Reflection is used to instantiate new objects, invoke methods, and to get and set fields at runtime. While reflection can be used for legitimate purposes, it is also commonly employed by malware to obfuscate malicious behavior.
Contains X509TrustManager that Accepts All Certificates	**Present**	The app contains a X509TrustManager that does not validate certificates. Any network connection that uses this trust manager is vulnerable to a man-in-the-middle by an attacker using a self-signed certificate. The following methods are found to have weak trust manager implementations: •io.fabric.sdk.android.services.network.PinningTrustManager.getAcceptedIssuers() OWASP: 2016-M3-Insecure Communication NIAP: FIA_X509_EXT.1.1

Creates Blowfish Key with Weak Length	**Not Present**	The app does not create a Blowfish key with less than 128 bits in length. OWASP: 2016-M5-Insufficient Cryptography
Creates RSA Keys with Weak Modulus Length	**Not Present**	The app does not create an RSA key with modulus length less than 1024 bits. OWASP: 2016-M5-Insufficient Cryptography NIAP: FCS_CKM.1.1(1)
Does not Update Security Provider	**Not Present**	The app uses the dynamic GmsCore_OpenSSL Provider to ensure that the device's security provider is always updated. OWASP: 2016-M1-Improper Platform Usage; 2016-M5-Insufficient Cryptography

Dynamically Loads Java Classes	**Present**	The app dynamically loads Java classes. If these classes are loaded from untrusted sources, such as a shared directory, the network, or an app from a different developer, it could be used by an attacker to gain code execution. Dynamic loading of Java code is found in the following methods: •androidx.appcompat.app.AppCompatViewInflater.createViewBy Prefix(Context, String, String) •androidx.appcompat.view.SupportMenuInflater$MenuState.newI nstance(String, Class, Object) •androidx.coordinatorlayout.widget.CoordinatorLayout.parseBeha vior(Context, AttributeSet, String) •androidx.core.app.AppComponentFactory.instantiateActivityCom pat(ClassLoader, String, Intent) •androidx.core.app.AppComponentFactory.instantiateApplication Compat(ClassLoader, String) •androidx.core.app.AppComponentFactory.instantiateProviderCo mpat(ClassLoader, String) •androidx.core.app.AppComponentFactory.instantiateReceiverCo mpat(ClassLoader, String, Intent) •androidx.core.app.AppComponentFactory.instantiateServiceCo mpat(ClassLoader, String, Intent) •androidx.fragment.app.Fragment.instantiate(Context, String, Bundle) •androidx.fragment.app.Fragment.isSupportFragmentClass(Cont ext, String) •androidx.preference.PreferenceInflater.createItem(String, String, AttributeSet) •androidx.recyclerview.widget.RecyclerView.createLayoutManag er(Context, String, AttributeSet, int, int) •androidx.transition.TransitionInflater.createCustom(AttributeSet, Class, String) •com.crashlytics.android.answers.AppMeasurementEventLogger. getClass(Context) •com.crashlytics.android.core.DefaultAppMeasurementEventList enerRegistrar.getClass(String) •io.fabric.sdk.android.services.common.FirebaseAppImpl.getInst ance(Context) OWASP: 2016-M7-Client Code Quality NIAP: FPT_TUD_EXT.1.4
Executes Environment Commands	**Not Present**	The app does not execute Linux-style environment commands. OWASP: 2016-M7-Client Code Quality

Insecure Pseudo-random Number Generation	**Present**	The app uses a psuedo-random number generator which returns a predictable sequence of numbers that is unsuitable for security purposes. The java.util.Random and java.lang.Math classes should not be used to generate random numbers for secure use. While SecureRandom is the correct class for this purpose, it is recommended to avoid seeding a SecureRandom object. In some implementations of SecureRandom, seeding it may completely replace the cryptographically strong default seed. Insecure psuedo-random number generators are found in the following methods: •androidx.transition.Explode.calculateOut(View, Rect, int) •com.amazon.device.ads.aftv.KSOServiceBinder.sendMessage() •com.crashlytics.android.answers.RandomBackoff.randomJitter() •com.crashlytics.android.core.CrashTest.stackOverflow() •com.xumo.xumo.service.XumoWebService$12.onResponse(JSONObject) •com.xumo.xumo.util.XumoUtil.getRandomNumber(int) OWASP: 2016-M5-Insufficient Cryptography NIAP: FCS_RBG_EXT.1.1
Logs Information	**Present**	The app prints logging information to the system log. While apps often log information for debugging purposes, this should generally be removed before an app is put into production. No sensitive information, such as keys or authentication tokens, should ever be written to the system log. OWASP: 2016-M2-Insecure Data Storage NIAP: FCS_CFG_EXT.1.2
Providers Accessible to Other Apps	**Not Present**	The app does not contain content providers, no content provider is exported, or access to all content providers is restricted by use of permissions. OWASP: 2016-M1-Improper Platform Usage NIAP: FMT_CFG_EXT.1.2
Receivers Accessible to Other Apps	**Not Present**	The app does not contain receivers, no receivers are exported, or access to all exported receivers is restricted by use of permissions. OWASP: 2016-M1-Improper Platform Usage
Requests Root Access	**Not Present**	The app does not request root access. OWASP: 2016-M8-Code Tampering
Services Accessible to Other Apps	**Not Present**	The app does not contain services, no services are exported, or access to all services is restricted by use of permissions. OWASP: 2016-M1-Improper Platform Usage

SMS CVE-2014-8610	**Not Present**	The app does not send text messages or has the required SMS permission. It is protected from vulnerability CVE-2014-8610, where an unprivileged app can resend all the SMS stored in the user's phone to their corresponding recipients or senders without user interaction. OWASP: 2016-M1-Improper Platform Usage
Source Code is not Obfuscated	**Present**	The app does not obfuscate the majority of its code by renaming classes, fields, methods, and variables. This allows an adversary or competitor to decompile the app into near-original source code. It is recommended to obfuscate the app's code using a tool such as ProGuard to make it more difficult to reverse engineer. OWASP: 2016-M9-Reverse Engineering
Uses Cipher That Does not Provide Integrity	**Not Present**	This app does not use a cipher that does not provide data integrity. OWASP: 2016-M5-Insufficient Cryptography
Uses Dangerous Permissions	**Not Present**	No dangerous permissions were requested by the app. OWASP: 2016-M1-Improper Platform Usage
Uses DES or 3DES Cipher	**Not Present**	The app does not use the DES or 3DES cipher. OWASP: 2016-M5-Insufficient Cryptography
Uses Electronic Code Book Mode Cipher Mode	**Not Present**	This app does not use ciphers with the Electronic Code Book (ECB) mode. OWASP: 2016-M5-Insufficient Cryptography
Uses MD5 Hashing Algorithm	**Present**	The app uses the weak MD5 hashing algorithm. The MD5 algorithm is dangerous if used for sensitive data because it highly vulnerable to collision attacks. OWASP: 2016-M5-Insufficient Cryptography
Uses NullCipher	**Not Present**	This app does not use NullCipher. OWASP: 2016-M5-Insufficient Cryptography

Uses Object Deserialization	**Present**	The app calls the java.io.ObjectInputStream.readObject() method to deserialize objects into memory. Object deserialization is a common source of vulnerabilities, particularly when the object may be from an untrusted source. It is recommended to avoid object deserialization when possible, or otherwise to harden the ObjectInputStream against attacks. One strong hardening technique is to override the resolveClass() to only allow expected classes. This can be implemented by subclassing ObjectInputStream. Object deserialization is found in the following methods: •androidx.versionedparcelable.VersionedParcel.readSerializable() OWASP: 2016-M7-Client Code Quality
Uses RSA Encryption Algorithm Without Padding	**Not Present**	The app does not use the RSA algorithm without padding. OWASP: 2016-M5-Insufficient Cryptography
Uses SHA1 Hashing Algorithm	**Present**	The app uses the weak SHA1 hashing algorithm. The SHA1 algorithm is dangerous if used for sensitive data because it vulnerable to collision attacks. OWASP: 2016-M5-Insufficient Cryptography

Weak Construction of Socket Factory	**Present**	The app creates a SSL socket factory that may be vulnerable to man-in-the-middle (MitM) attacks. The SSLSocketFactory class creates sockets that do not automatically perform certificate hostname validation, leaving the burden on the developer to manually create and use a HostnameVerifier. Failure to do so could allow an attacker to MitM the SSL socket connection by presenting legitimate signed certificate for a different hostname. The SSLCertificateSocketFactory class creates sockets that automatically perform hostname validation, but only when instantiated with 'String host'. SSLCertificateSocketFactory also has a method 'getInsecure' that returns a SSL socket factory with all security checks disabled, which would allow an attacker to perform a MitM attack with any certificate. Google recommends choosing the highest level networking API possible, such as HttpsURLConnection, because the higher level APIs perform these security checks automatically. The following methods create SSL socket factories that either do not automatically perform hostname validation or do not automatically perform any security validation: •com.google.android.gms.measurement.internal.zzfv.createSocket() •com.google.android.gms.measurement.internal.zzfv.createSocket(InetAddress, int) •com.google.android.gms.measurement.internal.zzfv.createSocket(InetAddress, int, InetAddress, int) •com.google.android.gms.measurement.internal.zzfv.createSocket(Socket, String, int, boolean) •com.google.android.gms.measurement.internal.zzfv.createSocket(String, int) •com.google.android.gms.measurement.internal.zzfv.createSocket(String, int, InetAddress, int) OWASP: 2016-M3-Insecure Communication
Weak RSA Modulus Length of App Signing Certificate	**Not Present**	The app is signed with a key of 2048 bit length or greater, as recommended by Google. The private signing key is protected. OWASP: 2016-M5-Insufficient Cryptography
Weakly Configured XML Parser	**Not Present**	No potential weakly configured XML parsing is found. OWASP: 2016-M7-Client Code Quality
Webview Contains JavaScript Interface	**Not Present**	The app does not expose an interface to access internal Java methods from JavaScript. OWASP: 2016-M7-Client Code Quality

Hard-coded Values Found

URLs	Country
http://crl.verisign.com/pca3.crl0)U	US
http://developer.android.com/tools/extras/support-library.html	US
http://fabric.io/terms/fabric	US
http://logo.verisign.com/vslogo.gif0U%0++04+(0&0$+0http://ocsp.verisign.com01U*0(0&$	*
http://ns.adobe.com/xap/1.0/	*
http://ns.adobe.com/xap/1.0/mm/	*
http://ns.adobe.com/xap/1.0/sType/ResourceRef#	*
http://schemas.android.com/aapt	*
http://schemas.android.com/apk/res-auto	*
http://schemas.android.com/apk/res/android	*
http://schemas.microsoft.com/DRM/2007/03/protocols/AcquireLicense	US
http://scripts.sil.org/OFL).http://scripts.sil.org/OFL	US
http://tools.android.com	US
http://www.apache.org/licenses/LICENSE-2.0	UA
http://www.cbsnews.com/news/live/	US
http://www.cbsnews.com/news/live/?c=24	US
http://www.dom.com/path?	US
http://www.w3.org/1999/02/22-rdf-syntax-ns#	US
http://www.w3.org/ns/ttml#parameter	US
https://android-tv-app.xumo.com/geo-check/index.html	US
https://app-measurement.com/a	US
https://developer.android.com/topic/libraries/architecture/index.html	US
https://docs.google.com/viewer?url=http://try.crashlytics.com/terms/terms-of-service.pdf	US
https://docs.google.com/viewer?url=https://fabric.io/answers-agreement.pdf	US
https://e.crashlytics.com/spi/v2/events	US
https://firebase.google.com/terms	US
https://github.com/chrisjenx/Calligraphy	US
https://github.com/emilsjolander/StickyListHeaders	US
https://github.com/google/Exoplayer/blob/release-v2/LICENSE	US
https://github.com/google/volley/blob/master/LICENSE	US
https://github.com/googleads/googleads-ima-android/blob/master/LICENSE	US

https://goo.gl/NAOOOI	US
https://google.github.io/ExoPlayer/faqs.html#what-do-player-is-accessed-on-the-wrong-thread-warnings-mean	*
https://image.xumo.com/v1/	US
https://image.xumo.com/v1/assets/asset/%s/%dx%d.png	US
https://image.xumo.com/v1/assets/asset/%s/480x300.jpeg	US
https://image.xumo.com/v1/channels/channel/%s/%dx%d.png?type=%s	US
https://image.xumo.com/v1/channels/channel/%s/%dx%d.png?type=color_onBlack	US
https://image.xumo.com/v1/channels/channel/%s/480x300.png?type=channelTile	US
https://image.xumo.com/v1/providers/provider/%s/120x90.png?type=color_onBlack	US
https://imasdk.googleapis.com/native/sdkloader/native_sdk_v3.html	US
https://imasdk.googleapis.com/native/sdkloader/native_sdk_v3_debug.html	US
https://pagead2.googlesyndication.com/pagead/gen_204?id=gmob-apps	US
https://pubads.g.doubleclick.net/gampad/ads?sz=640x480&iu=/124319096/external/ad_rule_samples&ciu_szs=300x250&ad_rule=1&impl=s&gdfp_req=1&env=vp&output=vmap&unviewed_position_start=1&cust_params=deployment%3Ddevsite%26sample_ar%3Dpremidpostpod&cmsid=496&vid=short_onecue&correlator=	US
https://pubads.g.doubleclick.net/gampad/ads?sz=640x480&iu=/124319096/external/single_ad_samples&ciu_szs=300x250&impl=s&gdfp_req=1&env=vp&output=vast&unviewed_position_start=1&cust_params=deployment%3Ddevsite%26sample_ct%3Dlinear&correlator=	US
https://saa.cbsi.com/b/ss/cbsicbsnewssite/0	US
https://settings.crashlytics.com/spi/v2/platforms/android/apps/%s/settings	US
https://som.cbsi.com/b/ss/cbsicbsnewssite,cbsicbsiall/1/5.4/REDIR	US
https://vizio-app.xumo.com/config/provider-genre-mapping-data.json	US
https://widevine-dash.ezdrm.com/proxy?pX=5FE38E	US
https://www.google.com	US
https://www.googleapis.com/auth/appstate	US
https://www.googleapis.com/auth/datastoremobile	US
https://www.googleapis.com/auth/drive	US
https://www.googleapis.com/auth/drive.appdata	US
https://www.googleapis.com/auth/drive.apps	US
https://www.googleapis.com/auth/drive.file	US
https://www.googleapis.com/auth/fitness.activity.read	US
https://www.googleapis.com/auth/fitness.activity.write	US
https://www.googleapis.com/auth/fitness.blood_glucose.read	US
https://www.googleapis.com/auth/fitness.blood_glucose.write	US

https://www.googleapis.com/auth/fitness.blood_pressure.read	US
https://www.googleapis.com/auth/fitness.blood_pressure.write	US
https://www.googleapis.com/auth/fitness.body.read	US
https://www.googleapis.com/auth/fitness.body.write	US
https://www.googleapis.com/auth/fitness.body_temperature.read	US
https://www.googleapis.com/auth/fitness.body_temperature.write	US
https://www.googleapis.com/auth/fitness.location.read	US
https://www.googleapis.com/auth/fitness.location.write	US
https://www.googleapis.com/auth/fitness.nutrition.read	US
https://www.googleapis.com/auth/fitness.nutrition.write	US
https://www.googleapis.com/auth/fitness.oxygen_saturation.read	US
https://www.googleapis.com/auth/fitness.oxygen_saturation.write	US
https://www.googleapis.com/auth/fitness.reproductive_health.read	US
https://www.googleapis.com/auth/fitness.reproductive_health.write	US
https://www.googleapis.com/auth/games	US
https://www.googleapis.com/auth/games.firstparty	US
https://www.googleapis.com/auth/games_lite	US
https://www.googleapis.com/auth/plus.login	US
https://www.googleapis.com/auth/plus.me	US
https://www.xumo.tv/channel/%s/%s?utm_source=android	US
https://www.xumo.tv/channel/%s/%s?v=%s&utm_source=android	US
https://www.xumo.tv/video/%s/%s?utm_source=android	US

IP Addresses	Country
1.2.10.27	US

Emails

hello@rfuenzalida.com

impallari@gmail.com

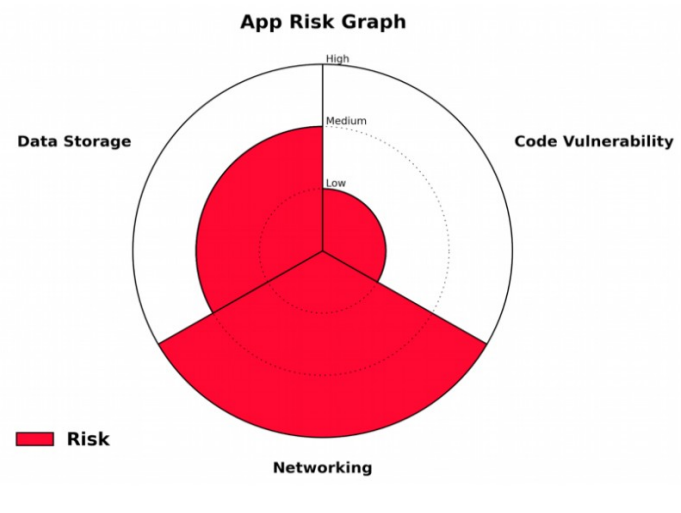
matt@pixelspread.com

support@xumo.com

Upgrade to App Vulnerability Assessment

Thank you for trying the free AppCritique scan! Data breaches lead to loss of intellectual property, litigation, customer dissatisfaction and loss in revenue. Many significant vulnerabilities cannot be detected by our Free Report. AppCritique offers the **App Vulnerability Assessment** (AVA) as a deep dive of your app conducted by the AppCritique experts. The AVA service includes:

- Expert analysis of your app with risk assessment write-ups
- Recommendations and remediations
- Q&A session between your app developers or assurance team and the expert AppCritique analysts
- AVA checks include:
  - **Code Vulnerabilities**
    - Inter-app communications
    - Components vulnerable to manipulation
    - Unauthenticated or unfiltered input
    - Local SQL injection
    - Unsafe native code
    - Dynamically loaded code
    - Hard-coded credentials
    - Deprecated cryptography
  - **Data Storage**
    - Data accessible in unencrypted backups
    - Publicly accessible sensitive information
    - Credentials outside secure store
    - Data privacy analysis
    - Side channel data leakage
  - **Networking**
    - Certificate validation issues
    - Unencrypted protocols
    - Weak endpoint encryption
    - Back-end analysis
    - Man-in-the-middle attack scenarios



Sample App Risk Graph

Contact Us

Email us at AppCritique@bah.com to set up your App Vulnerability Assessment!